

Samson, Harvey S.

2018 – 00542

Exercise 6: Networking

- Listing networks (adding ping container does not show in 'inspect bridge')

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.18363.1556]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
d2b0b2798201        bridge             bridge             local
7f9e0e321e44        host              host              local
d57147911ddb        none              null              local

C:\WINDOWS\system32>docker network inspect bridge
[
  {
    "Name": "bridge",
    "Id": "d2b0b2798201a200f5c3499278b13f0578e62e8ab487e05b7b26fed59e50775a",
    "Created": "2021-05-30T08:52:55.5612182Z",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": null,
      "Config": [
        {
          "Subnet": "172.17.0.0/16",
          "Gateway": "172.17.0.1"
        }
      ]
    },
    "Internal": false,
    "Attachable": false,
    "Ingress": false,
    "ConfigFrom": {
      "Network": ""
    },
    "ConfigOnly": false,
    "Containers": {},
    "Options": {
      "com.docker.network.bridge.default_bridge": "true",
      "com.docker.network.bridge.enable_icc": "true",
      "com.docker.network.bridge.enable_ip_masquerade": "true",
      "com.docker.network.bridge.host_binding_ipv4": "0.0.0.0",
      "com.docker.network.bridge.name": "docker0",
      "com.docker.network.driver.mtu": "1500"
    },
    "Labels": {}
  }
]
```

```
Administrator: Command Prompt

]
}

C:\WINDOWS\system32>docker run --rm -d --name dummy harveysamson/ping:1.0
801ca00fec583fe566ecf791555979a6ecc4e54dc735d8784d89b9e49b268409

C:\WINDOWS\system32>docker network inspect bridge
[
  {
    "Name": "bridge",
    "Id": "1b8a4816076c6fe580d533d14e2948401c0d8b3b330e4080feac0b89c6d8cb37",
    "Created": "2021-05-30T10:39:02.9985685Z",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": null,
      "Config": [
        {
          "Subnet": "172.17.0.0/16",
          "Gateway": "172.17.0.1"
        }
      ]
    },
    "Internal": false,
    "Attachable": false,
    "Ingress": false,
    "ConfigFrom": {
      "Network": ""
    },
    "ConfigOnly": false,
    "Containers": {},
    "Options": {
      "com.docker.network.bridge.default_bridge": "true",
      "com.docker.network.bridge.enable_icc": "true",
      "com.docker.network.bridge.enable_ip_masquerade": "true",
      "com.docker.network.bridge.host_binding_ipv4": "0.0.0.0",
      "com.docker.network.bridge.name": "docker0",
      "com.docker.network.driver.mtu": "1500"
    },
    "Labels": {}
  }
]
```

```
Administrator: Command Prompt

"Driver": "bridge",
"EnableIPv6": false,
"IPAM": {
  "Driver": "default",
  "Options": null,
  "Config": [
    {
      "Subnet": "172.17.0.0/16",
      "Gateway": "172.17.0.1"
    }
  ]
},
"Internal": false,
"Attachable": false,
"Ingress": false,
"ConfigFrom": {
  "Network": ""
},
"ConfigOnly": false,
"Containers": {},
"Options": {
  "com.docker.network.bridge.default_bridge": "true",
  "com.docker.network.bridge.enable_icc": "true",
  "com.docker.network.bridge.enable_ip_masquerade": "true",
  "com.docker.network.bridge.host_binding_ipv4": "0.0.0.0",
  "com.docker.network.bridge.name": "docker0",
  "com.docker.network.driver.mtu": "1500"
},
"Labels": {}
}
]

C:\WINDOWS\system32>docker run --rm -d -e PING_TARGET=172.17.0.2 --name pinger harveysamson/ping:1.0
0328c2f2d78019cea4f50c8068f803d202a2dea94966693e333abb4ad644432b

C:\WINDOWS\system32>docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
0328c2f2d780        harveysamson/ping:1.0  "/bin/bash"        24 seconds ago     Up 14 seconds      0.0.0.0:80->0.0.0.0:80  pinger

C:\WINDOWS\system32>docker logs pinger
Error: No such container: 0328c2f2d78019cea4f50c8068f803d202a2dea94966693e333abb4ad644432b

C:\WINDOWS\system32>docker stop dummy
Error response from daemon: No such container: dummy

C:\WINDOWS\system32>
```

- Managing custom networks (same issue as listing networks)

```
Administrator: Command Prompt

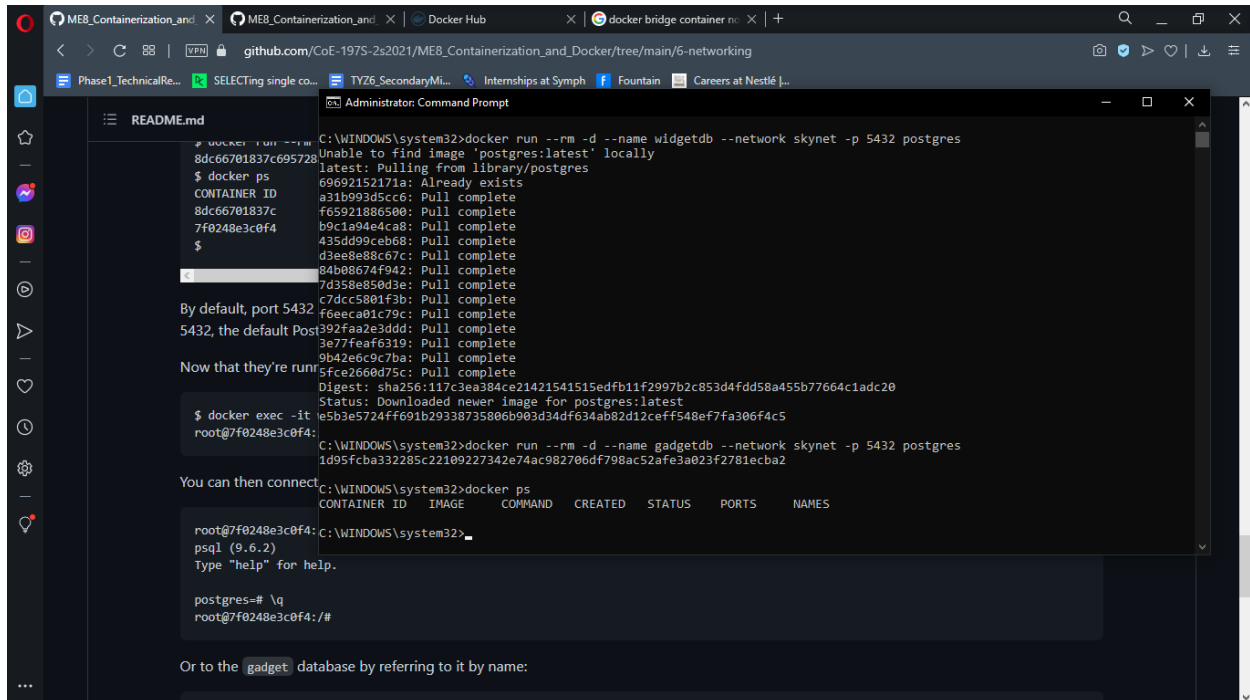
C:\WINDOWS\system32>docker network create skynet
15b8370a221e56426af773553a1a7eb69379aaf85065436a95d653314a5f254b

C:\WINDOWS\system32>docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
1b8a4816076c        bridge              bridge              local
7f9e0e321e44        host                host                local
d57147911dbb        none                null                local
15b8370a221e        skynet              bridge              local

C:\WINDOWS\system32>docker network inspect skynet
[
  {
    "Name": "skynet",
    "Id": "15b8370a221e56426af773553a1a7eb69379aaf85065436a95d653314a5f254b",
    "Created": "2021-05-30T10:43:39.3538108Z",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": {},
      "Config": [
        {
          "Subnet": "172.18.0.0/16",
          "Gateway": "172.18.0.1"
        }
      ]
    },
    "Internal": false,
    "Attachable": false,
    "Ingress": false,
    "ConfigFrom": {
      "Network": ""
    },
    "ConfigOnly": false,
    "Containers": {},
    "Options": {},
    "Labels": {}
  }
]

C:\WINDOWS\system32>
```

- Connecting between containers in a network (docker ps does not show anything)



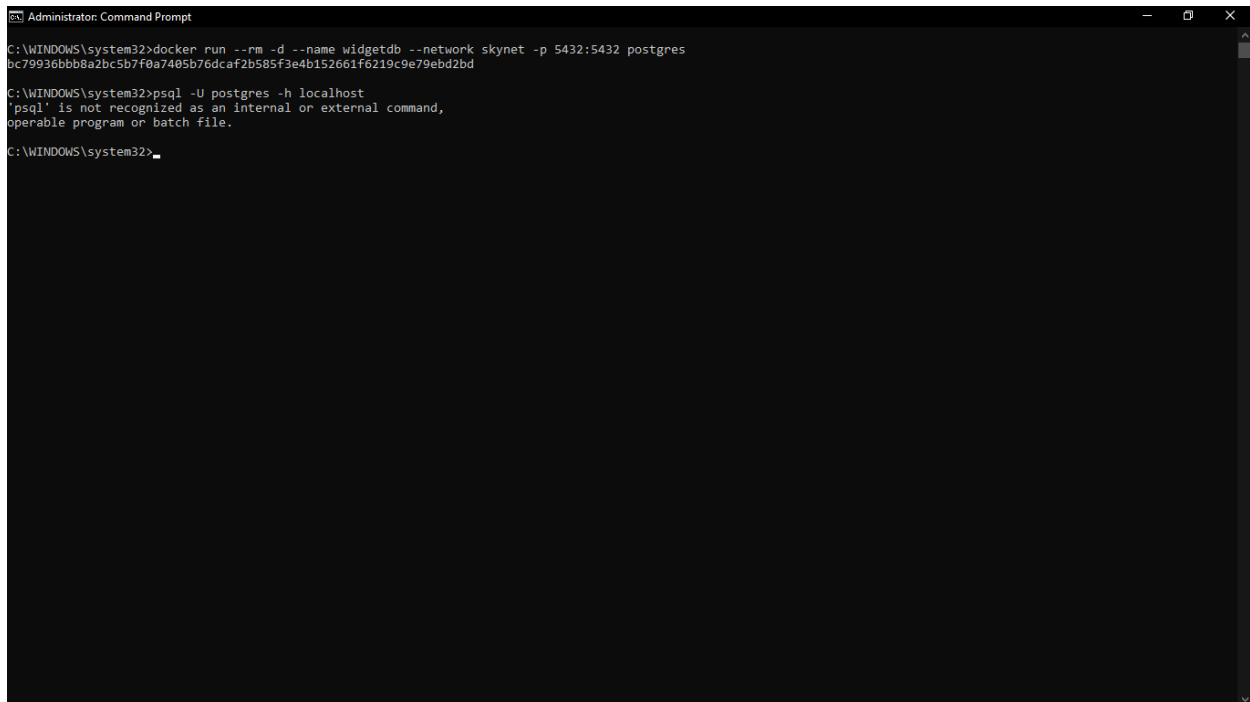
The screenshot shows a web browser window displaying a GitHub page for a Docker network setup. Below the browser, a Windows Command Prompt window is open, showing the following commands and output:

```
C:\WINDOWS\system32>docker run --rm -d --name widgetdb --network skynet -p 5432 postgres
Unable to find image 'postgres:latest' locally
latest: Pulling from library/postgres
69692152171a: Already exists
a31b993d5cc6: Pull complete
8dc66701837c: Pull complete
f65921886500: Pull complete
b9c1a94e4ca8: Pull complete
b9c1a94e4ca8: Pull complete
435dd99ceb68: Pull complete
d3ee8e88c67c: Pull complete
84b88674f942: Pull complete
7d358e850d3e: Pull complete
c7dccc5801f3b: Pull complete
By default, port 5432
5432, the default Postgres
3e77feaf6319: Pull complete
9b42e6c9c7ba: Pull complete
Now that they're running:
Digest: sha256:117c3ea384ce21421541515edfb11f2997b2c853d4fdd58a455b77664c1adc20
Status: Downloaded newer image for postgres:latest
e5b3e5724ff691b29338735806b903d34df634ab82d12ceff548ef7fa306f4c5
root@7f0248e3c0f4: C:\WINDOWS\system32>docker run --rm -d --name gadgetdb --network skynet -p 5432 postgres
1d95fcb332285c22109227342e74ac982706df798ac52afe3a023f2781ecba2

You can then connect
C:\WINDOWS\system32>docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
root@7f0248e3c0f4: C:\WINDOWS\system32>
psql (9.6.2)
Type "help" for help.

postgres=# \q
root@7f0248e3c0f4: /#
```

Or to the `gadget` database by referring to it by name:



The screenshot shows a Windows Command Prompt window with the following commands and output:

```
C:\WINDOWS\system32>docker run --rm -d --name widgetdb --network skynet -p 5432:5432 postgres
bc79936bbb8a2bc5b7f0a7405b76dcdf2b585f3e4b152661f6219c9e79ebd2bd

C:\WINDOWS\system32>psql -U postgres -h localhost
'psql' is not recognized as an internal or external command,
operable program or batch file.

C:\WINDOWS\system32>
```